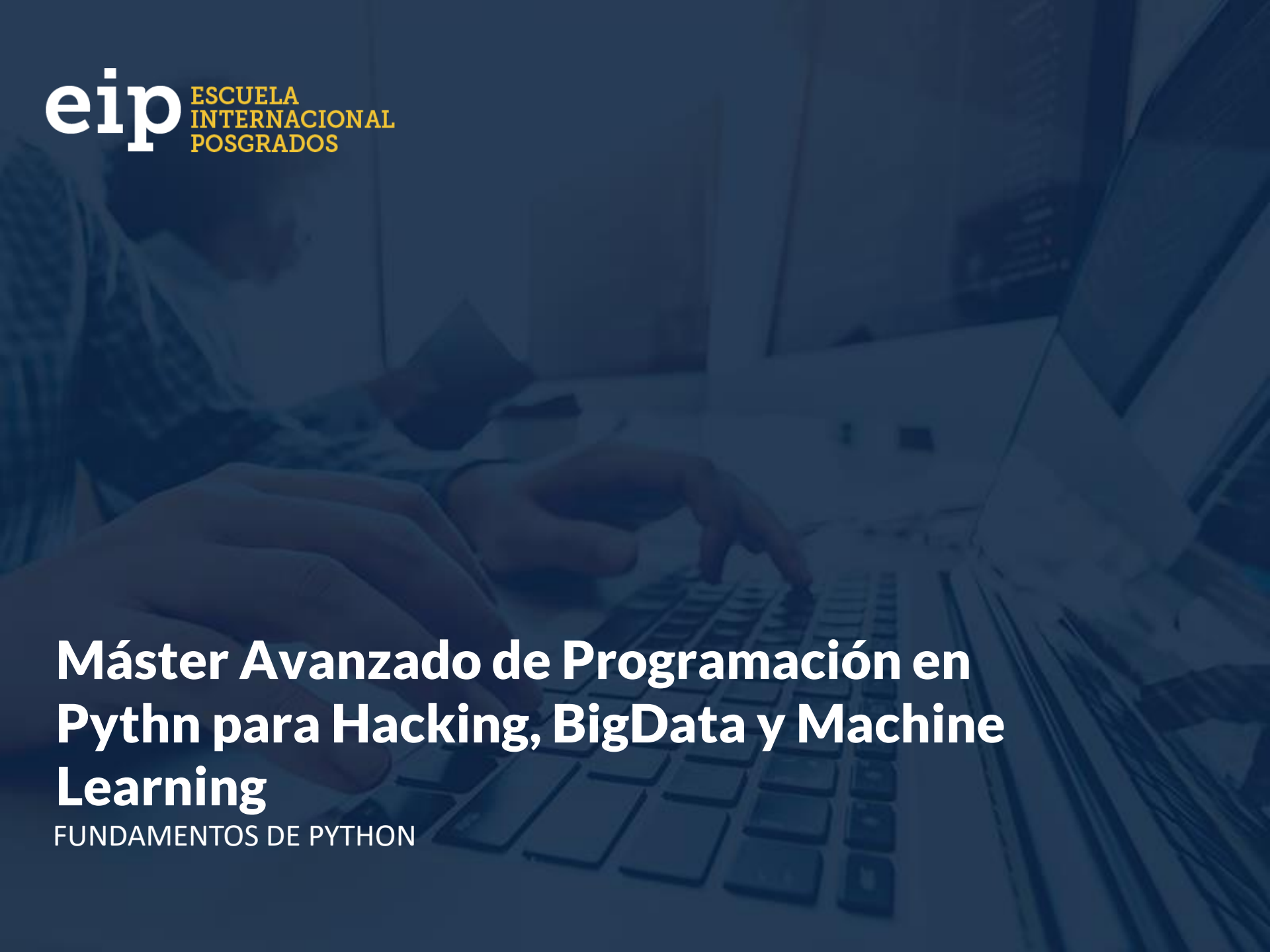


eip ESCUELA INTERNACIONAL POSGRADOS

Máster Avanzado de Programación en Pythn para Hacking, BigData y Machine Learning

FUNDAMENTOS DE PYTHON

LECCIÓN 01

HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN. FILOSOFÍA DE PTYHON.

ÍNDICE

INTRODUCCIÓN

En esta lección haremos un repaso por la historia de los diferentes lenguajes de programación, analizando cómo han evolucionado para adaptarse a la necesidad de cada época.

Estudiaremos desde los primeros lenguajes, donde se utilizaban tarjetas perforadas hasta los lenguajes de programación más recientes, como Python, Java o C++.

OBJETIVOS

Al finalizar esta lección serás capaz de:

- Conocer cuáles fueron los orígenes de los lenguajes de programación y su historia.
- Conocer las diferentes generaciones de lenguajes de programación.
- Comprender la tipología de los lenguajes de programación.
- Conocer la historia y filosofía de Python.

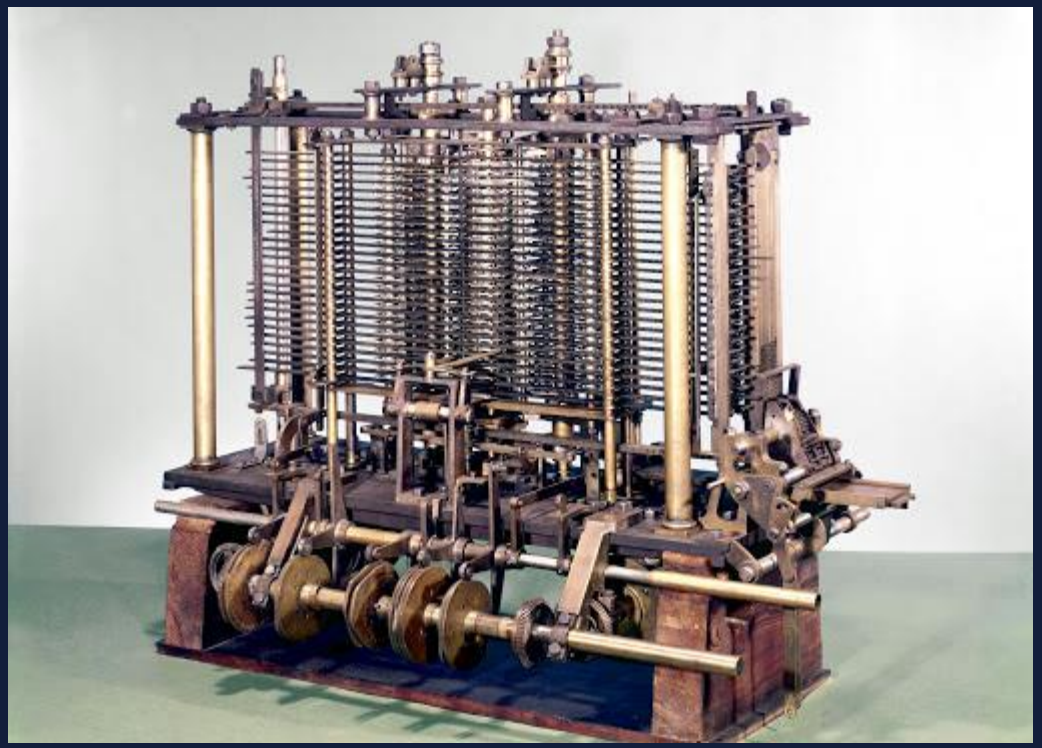
Introducción a los lenguajes de programación

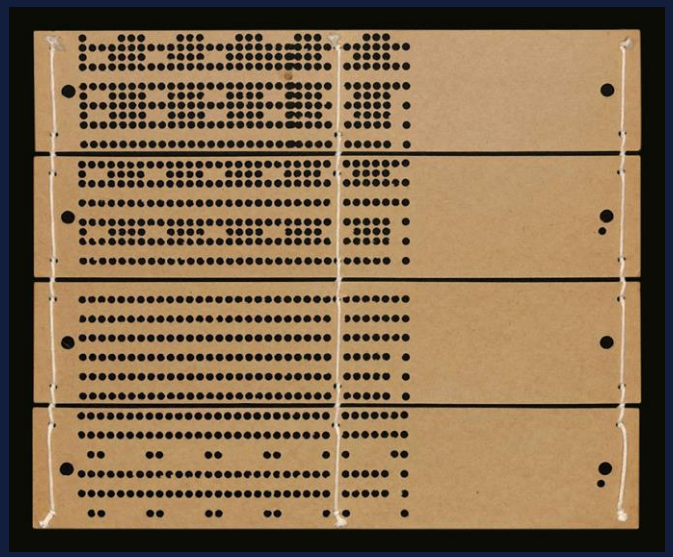
- A lo largo de la historia se ha perseguido el desarrollo de máquinas que ayudaran a realizar tareas mecánicas y tediosas de una forma más sencilla.

- Esta historia va desde el siglo XVII y el desarrollo de las primeras máquinas calculadoras mecánicas hasta las computadoras modernas que conocemos en la actualidad.

- El principal descubrimiento que cambió la forma de comprender las máquinas fue la **máquina analítica**, diseñada por **Charles Babage** en 1840.

- La **máquina analítica** fue la **primera máquina programable.**
- Permitía que los cálculos realizaran saltos condicionales y bucles.





- Un siglo después, **Alan Turing** definió de forma abstracta un ordenador como una máquina capaz de leer y escribir ceros y unos en una cinta infinita. Esta máquina se recibió el nombre de **máquina de Turing.**

- La máquina de Turing, se movía escribiendo y leyendo a lo largo de la cinta siguiendo una secuencia de instrucciones.

- **El primer lenguaje de programación surgió en 1843 de mano de Ada Lovelace. Fue la primera persona en implementar un algoritmo para a máquina analítica.**

- **Gracias a esta contribución, se ganó el título de primera programadora el mundo.**

- En 1945 se sentaron las bases del primer computador eléctrico digital, gracias a la arquitectura Von Neumann (en honor a su creador **John Von Neumann**).

- Se definen dos ideas claves:
  - Un programa que puede ser almacenado en memoria.
  - Un conjunto de instrucciones que permitan cargar y ejecutar un programa.

Gracias a este descubrimiento, en 1948 se construyó en la Universidad de Manchester el primer ordenador eléctrico digital.

- A partir de estos avances, surgió la necesidad de desarrollar programas diseñados a resolver tareas específicas (lo que hoy en día conocemos como **software**).

- Estos programas son escritos en un determinado lenguaje de programación.

Un lenguaje de programación es un vocabulario y un conjunto de reglas gramaticales para dar instrucciones a una máquina para que ésta realice un conjunto de tareas específicas.

- El código máquina presenta los siguientes inconvenientes:
    - Lejano del lenguaje natural.
    - Sintaxis muy estricta.
    - Operaciones muy limitadas y elementales.
    - Dependiente del hardware.
- **SOLUCIÓN**: desarrollo de los lenguajes de programación de alto nivel:
    - Interpretables.
    - Independiente de la máquina que lo utilice.
    - Fácil de depurar.
    - Portabilidad a otras plataformas.

- **PRIMERA** generación de los lenguajes de programación:
  - Lenguajes a nivel de máquina. Se comunicaban por medio de ceros y unos.
  - **INCONVENIENTES**: escribir un programa en código máquina es una tarea ardua y además está sujeto a errores.
  - **VENTAJAS**: alta velocidad de ejecución, ya que se ejecutan directamente en la CPU del ordenador. NO necesitan de un compilador para convertir el lenguaje de programación a código máquina.

- **SEGUNDA** generación de los lenguajes de programación:
  - Los lenguajes de segunda generación hacen referencia al lenguaje ensamblador.
  - Lenguajes específicos para un determinado hardware.
  - Son más interpretables que los lenguajes de primera generación.
  - Siguen utilizándose hoy en día, aunque los lenguajes de alto nivel los han reemplazado.

- Lenguajes de TERCERA generación:
  - Lenguajes de propósito general.
  - Utilizados en ámbitos científicos y empresariales.
  - Son muy interpretables.
  - Independientes del hardware que lo ejecute.
  - Fáciles de depurar y corregir errores

Ejemplos de lenguajes de tercera generación:

- **FORTRAN**
- **PASCAL**
- **C**
- **C++**
- **JAVA**

- Los primeros lenguajes de programación de alto nivel surgieron a finales de la década de los 50, y fueron **FORTRAN** en 1956 y **Lisp** en 1958.

- FORTRAN: desarrollado por **John Backus** y se le considera como el lenguaje de programación más antiguo que sigue utilizándose hoy en día. Entre sus características podemos destacar que es un **lenguaje imperativo** y **compilado**

Example: It is required to compute the following quantities

$$P_i = \sqrt{\sin^2(A_i B_i + C_i) + \cos^2(A_i B_i - C_i)}$$

$$Q_i = \sin^2(A_i + C_i) + \cos^2(A_i - C_i)$$

for i = 1, ..., 100. A possible FORTRAN program for this calculation follows.

C ← FOR COMMENT / STATEMENT NUMBER (1–5)	CONTINUATION (6)	FORTRAN STATEMENT (7–72)
1		TRIGF(X,Y) = SINF (X+Y)**2+COSF(X-Y)**2
2		DIMENSION A(100), B(100), C(100), P(100), Q(100)
3		READ 8, A, B, C
4		DO 6 I = 1,100
5		P(I) = SQRTF(TRIGF(A(I)*B(I), C(I)))
6		Q(I) = TRIGF(A(I), C(I))
7		PRINT 8, (A(I), B(I), C(I), P(I), Q(I), I = 1,100)
8		FORMAT (5F 10.4)
9		STOP

- LISP: desarrollado por **John McCarty** en el Instituto de Tecnología de Massachusetts (MIT).

- Fue propuesto para desarrollar labores de **Inteligencia Artificial** y es uno de los lenguajes de programación más antiguos que siguen utilizándose y que pueden ser utilizados en lugar de Ruby o Python.

- Es un lenguaje funcional e interpretado.

3. The Function LENGTH, defined Recursively

If we want to define LENGTH recursively, the expression is even simpler and easier:

To find the LENGTH of a list M:

If the list is null, take 0. Else,
Add 1 to the length of CDR M.

The LISP defining expression is:

```
(LENGTH (LAMBDA (M) (COND
0       1       2 2 2
     ((NULL M) 0)
     34       4  3

     (T (PLUS 1 (LENGTH (CDR M))))))
     3  4       5       6     6543210
```

- **CUARTA** generación de los lenguajes de programación:
  - Diseñados para que el programador definiera QUÉ tenían que hacer, en lugar de CÓMO hacerlo.
  - Orientados a reducir el esfuerzo del programador.
  - Utilizados comúnmente en programación de bases de datos y desarrollo de scripts.
  - INCONVENIENTES:
    - Son más ineficientes que los lenguajes de programación de tercera generación
  - Ejemplos: Perl, PHP, Python, Ruby, SQL

- Lenguajes de programación de **QUINTA** generación:
  - Diseñados para resolver problemas utilizando restricciones.
  - Tratan de hacer que el propio computador resuelva el problema, en lugar de que el programador le indique cómo hacerlo.
  - Utilizados en Inteligencia Artificial.
  - De ámbito académico.

Ejemplos de lenguajes de quinta generación:

- **Prolog**
- **Mercury**
- **OPS5**

Tipos de lenguajes de programación

- Desde 1954 hasta la actualidad se han registrado más de **2500 lenguajes de programación** diferentes.
- Analizaremos las características y funcionalidades que comparten los diferentes lenguajes de programación.
- Podemos identificar dos tipologías de lenguajes de programación:
  - Lenguajes de programación **interpretados**.
  - Lenguajes de programación **compilados**.

Lenguajes de programación procedurales

- Utilizados para ejecutar una secuencia de declaraciones que conducen a un resultado.
- Utilizan variables, bucles y otros elementos que los diferencian de los lenguajes de programación funcionales.
- Ventajas:
  - Mayor comprensión del código => facilita las tareas del diseño, depuración y mantenimiento.
  - Los códigos implementados bajo esta modalidad pueden ser compilados y almacenados en una librería.

Lenguajes de programación funcionales

- La programación funcional es un paradigma de le programación declarativa basado en el uso de funciones matemáticas.
- Se componen únicamente por definiciones de funciones, entendiéndolas como funciones matemáticas y no como subprogramas encapsulados en una función.
- En los problemas donde se abordan tareas matemáticas y algoritmos complejos los lenguajes de programación funcional ocupan una posición estratégica en la informática.

Lenguajes de programación orientados a objetos

- Es un paradigma de la programación basado en el que introduce el concepto de objetos.
- Estos objetos contienen datos en forma de campos (también conocido como atributos y propiedades) y código, también conocido como métodos.
- Los programas son diseñados de modo que los objetos puedan interactuar unos con otros, pudiendo acceder y modificar (a veces) los valores de sus propios campos o los de otros objetos.

Lenguajes de scripting

- Secuencia de comandos que suelen utilizarse para automatizar tareas repetitivas, hacer procesamiento de lotes e interactuar con el sistema operativo.

- El lenguaje de programación scripting no se compila, sino que hace uso de un **intérprete** que se encarga de leer y traducir cada línea del lenguaje de programación a medida que va siendo ejecutado.

Lenguajes de programación compilados

- Un lenguaje de programación compilado hace referencia a aquellos lenguajes que utilizan un compilador para **traducir** el **código** escrito por el usuario **a código máquina.**
- **Compilador** no es más que un programa que se encarga de **traducir todo el código a código máquina**, una vez hecho esto, el programa puede ser ejecutado.

Lenguajes de programación compilados

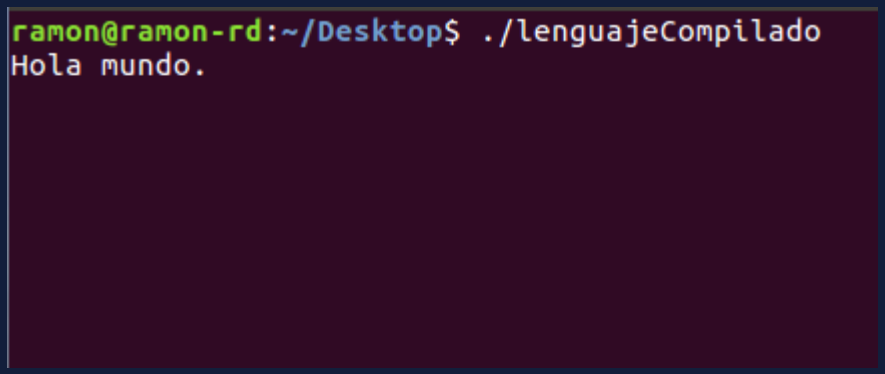
```
#include<iostream>

using namespace std;

int main(){

   cout << "Hola mundo." << endl;

}
```

Este es un programa sencillo que muestra por pantalla el famoso mensaje “Hola mundo”.

Para poder ejecutar este programa es necesario utilizar un compilador de C++, como g++.

Lenguajes de programación compilados

Tras compilar este programa, se nos crean dos nuevos ficheros: un fichero con la extensión ".o", que se refiere al código máquina que ha generado el compilador y otro fichero ejecutable que será el que utilicemos para ejecutar nuestro programa.



Lenguajes de programación interpretados

- Los lenguajes de programación interpretados no se hace uso de un compilador, si no que se utiliza un **intérprete**.

- La principal diferencia radica en que un lenguaje interpretado es convertido a código máquina a medida que éste es ejecutado.

Lenguajes de programación interpretados

- Un ejemplo de lenguaje interpretado sería el lenguaje de programación **Python.**
- El propio intérprete de Python se encarga de transformar el programa a medida que va ejecutándolo.

```
print("Hola mundo")
```

Lenguajes de programación interpretados

```
ramon@ramon-rd:~/Desktop$ python lenguajeInterpretado.py
Hola mundo
ramon@ramon-rd:~/Desktop$
```

En términos de diseño, implementación y prueba de un algoritmo, un lenguaje de programación interpretado es más rápido, en comparación con un lenguaje de programación compilado.

No obstante, un lenguaje compilado es mucho más rápido que uno interpretado. Esto se debe a que el programa que se ejecuta como resultado de compilar un código, ya se encuentra traducido a código máquina

Historia y filosofía de Python

- Python fue creado por Guido Van Rossum en el centro de las Matemáticas y la Informática a finales de 1980.
- El nombre asociado a este lenguaje de programación proviene por la afición de Guido por los humoristas británicos Monty Python.
- Python se ha establecido como uno de los lenguajes de programación optimizamos y bien establecidos como C, C++ y Java principalmente por su filosofía

Historia y filosofía de Python

- Python fue diseñado con la filosofía de que las tareas complejas puedan ser realizadas de forma sencilla.
- Python aprovechó el poder del movimiento de código abierto, lo que le ayudó a atraer una gran cantidad de usuarios de diversas especialidades.
- Al ser de código abierto, Python permitió a los desarrolladores crear pequeños programas y compartirlos entre sí con facilidad.

CARACTERÍSTICAS DE PYTHON I

- Es un lenguaje de programación interpretado y de alto nivel.
- Es un lenguaje de programación orientado a objetos.
- Muy versátil, ya que los programadores han desarrollado módulos con el objetivo de facilitar la resolución de trabajos en su área de especialización (cálculo científico, estadística, criptografía, bases de datos, etc.)

CARACTERÍSTICAS DE PYTHON II

- Diseño minimalista y legible.
- Ha sido portado a una gran variedad de plataformas, entre las que destacan Linux, Windows, Macintosh o Solaris.
- Python fue desarrollado para ser extensible lo que implica que sus usuarios pueden optar por utilizar una determinada funcionalidad atendiendo a sus necesidades.

CONCLUSIONES

- A lo largo de la historia han surgido más de 2500 lenguajes de programación para facilitar la vida de los programadores.
- Cada lenguaje tiene unas características que se adaptan al problema a resolver.
- Seleccionar qué lenguaje de programación atendiendo al problema a resolver es un factor clave.

MUCHAS GRACIAS POR SU ATENCIÓN

eip ESCUELA INTERNACIONAL POSGRADOS

ramonruedadelgado@gmail.com

Ramón Rueda Delgado
https://www.linkedin.com/in/ramon-rueda/

twitter.com/eiposgrados

facebook.com/eiposgrados

instagram.com/eiposgrados